

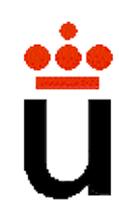


Titulación: Ingeniería Informática  
Asignatura: Fundamentos de  
Computadores

**Bloque 3:** Sistemas  
secuenciales

**Tema 9:** Módulos secuenciales  
básicos

Pablo Huerta Pellitero  
Luis Rincón Córcoles



# ÍNDICE

- Bibliografía
- Registros
  - Carga paralela
  - Registros de desplazamiento
- Contadores
  - Contadores síncronos
  - Contadores asíncronos
- Banco de registros



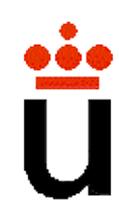
# BIBLIOGRAFÍA

- Román Hermida, Ana M<sup>o</sup> del Corral, Enric Pastor, Fermín Sánchez  
**“Fundamentos de Computadores”** , cap 5  
Editorial Síntesis
- Thomas L. Floyd  
**“Fundamentos de Sistemas Digitales”**, cap 8,9  
Editorial Prentice Hall
- Daniel D. Gajski  
**“Principios de Diseño Digital”**, cap 7  
Editorial Prentice Hall
- M. Morris Mano  
**“Diseño Digital”**, cap 7  
Editorial Prentice Hall



# REGISTROS

- Se denomina registro a un conjunto de biestables (típicamente *flip-flops*) que funcionan al unísono, compartiendo las señales de control (reloj, *reset*, etc).
- Las entradas y las salidas de datos del registro pueden ser en paralelo o en serie.
- Algunas de las entradas de control que pueden tener los registros son:
  - *Habilitación/inhabilitación del reloj*: se utiliza para controlar el funcionamiento del registro. Si el reloj está inhabilitado, el registro no cambiará su valor cuando llegue un flanco de reloj.
  - *Habilitación de entrada*: hace que la entrada llegue a los *flip-flops*, o que no llegue y por tanto mantengan su valor anterior. El efecto que tiene es equivalente a la *habilitación/inhabilitación del reloj*.
  - *Clear o reset*: entrada asíncrona común a todos los biestables que hace que todos pongan su salida a '0'.
  - *Preset o set*: entrada asíncrona común a todos los biestables que hace que todos pongan su salida a '1'.
  - *Habilitación de salida*: conecta/desconecta la salida del registro.



# ÍNDICE

- Bibliografía
- Registros
  - **Carga paralela**
    - Registros de desplazamiento
- Contadores
  - Contadores síncronos
  - Contadores asíncronos
- Banco de registros



# REGISTROS CON CARGA PARALELA

- Son los más comunes, y se utilizan para almacenamiento de datos. Tienen 'n' bits de entrada de datos y 'n' bits de salida de datos. Internamente hay 'n' *flip-flops*, y cada uno de ellos almacena uno de los bits del registro.
- Ejemplo: registro de 4 bits con carga paralela.

Símbolo

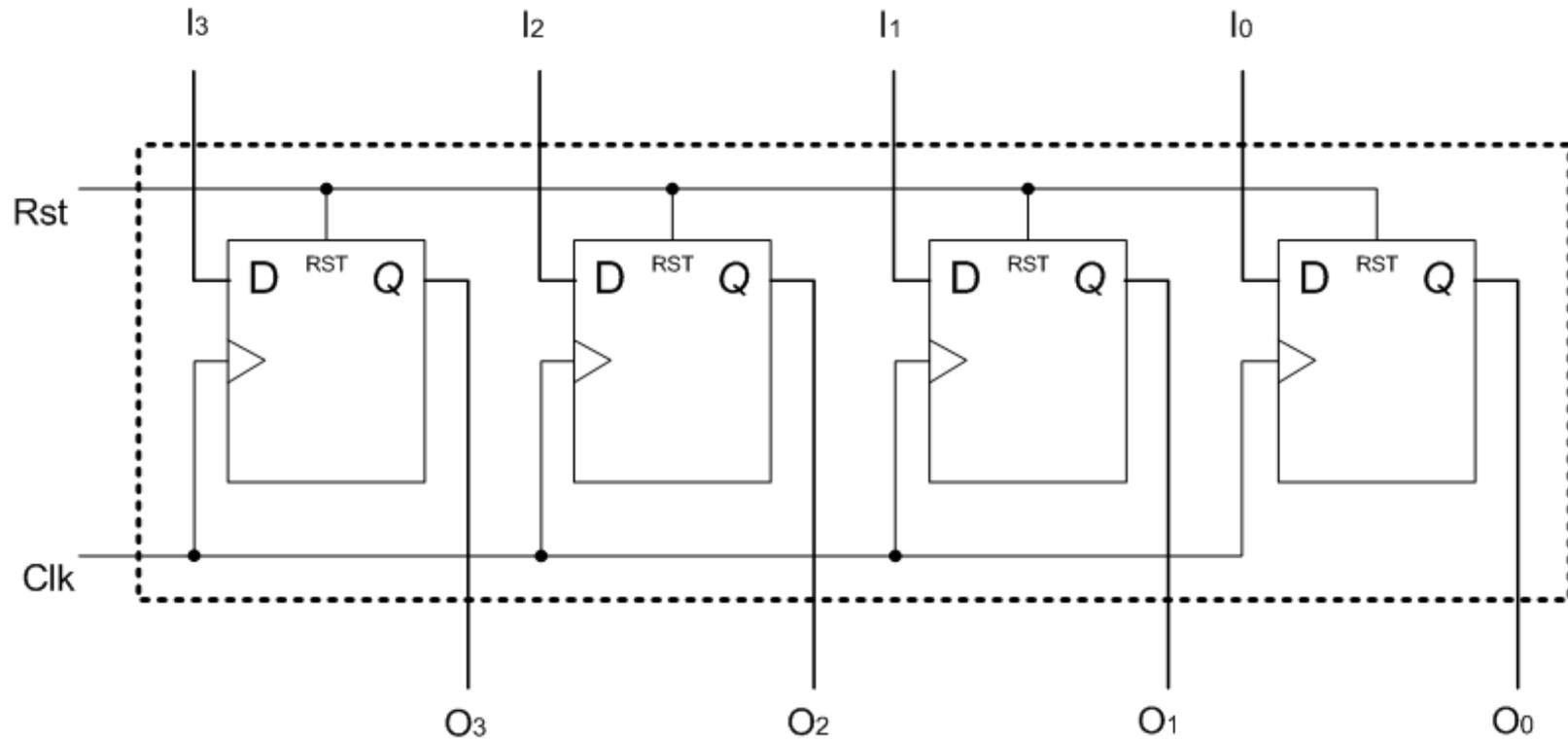
Tabla de funcionamiento

Rst	Clk	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
1	X	0	0	0	0
0	↑	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>



# REGISTROS CON CARGA PARALELA

- Registro con carga paralela:





# REGISTRO CON CARGA PARALELA: HABILITACIÓN DE RELOJ

- El registro con carga paralela puede tener una entrada de habilitación de reloj CE (*Clock Enable*). Si CE = '0', el reloj que llega a los biestables siempre vale '0'. Si CE = '1' a los biestables les llega la entrada de reloj Clk.
  - Esta forma de controlar el reloj se conoce como "*clock gating*". Puede dar problemas de sesgo ("*skew*") en el reloj del sistema, por lo que es preferible utilizar habilitación de entrada. Símbolo

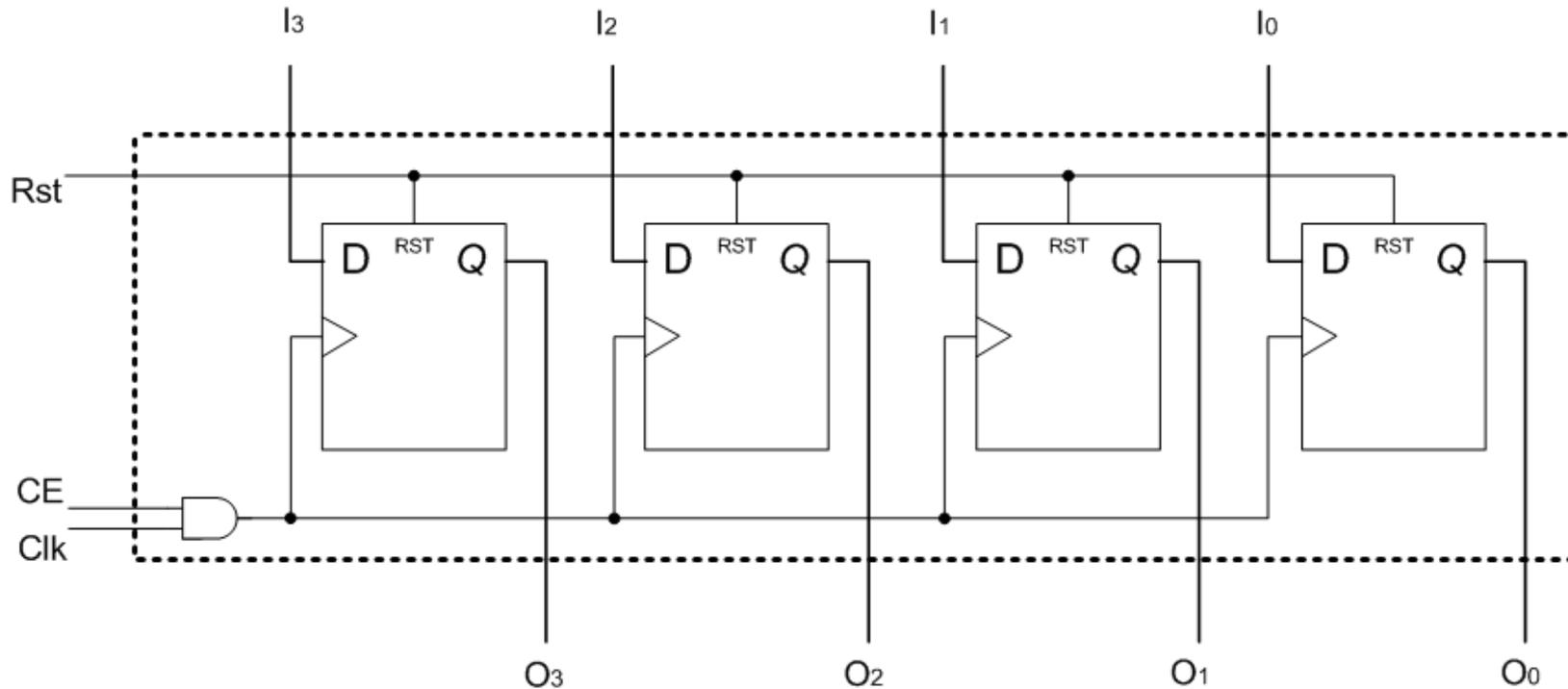
Tabla de funcionamiento

Rst	Clk	CE	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
1	X	X	0	0	0	0
0	↑	0	no cambia			
0	↑	1	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>



# REGISTROS CON CARGA PARALELA: HABILITACIÓN DE RELOJ

- Registro con carga en paralelo y habilitación de reloj:





# REGISTROS CON CARGA PARALELA: HABILITACIÓN DE ENTRADA

- La habilitación de entrada se puede hacer utilizando multiplexores a la entrada de los *flip-flops*. Estos multiplexores hacen que al *flip-flop* llegue o bien la entrada del registro o el valor que ya había almacenado en el flip-flop.

Símbolo

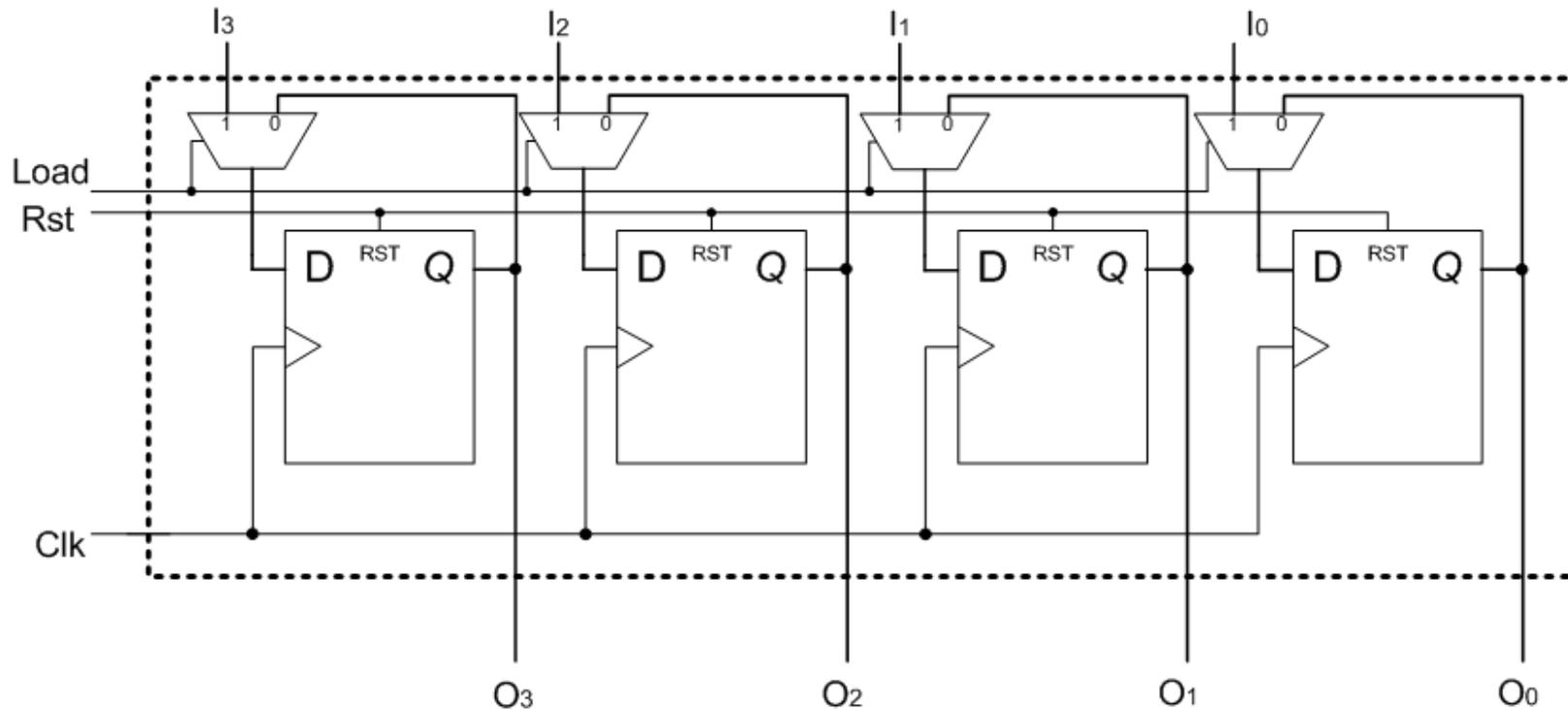
Tabla de  
funcionamiento

Rst	Clk	Load	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
1	X	X	0	0	0	0
0	↑	0	no cambia			
0	↑	1	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>



# REGISTROS CON CARGA PARALELA: HABILITACIÓN DE ENTRADA

- Registro con carga paralela y habilitación de entrada:





# REGISTROS CON CARGA PARALELA: HABILITACIÓN DE SALIDA

- La habilitación de salida permite, mediante un señal de control (OE, *Output Enable*), que el registro coloque en la salida el valor que tiene almacenado en los *flip-flops*, o que deje la salida en alta impedancia.
- Se implementa utilizando *buffers* triestado.

Símbolo

Tabla de funcionamiento: escritura

Rst	Clk	Load	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	X	X	0	0	0	0
0	↑	0	no cambia			
0	↑	1	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>

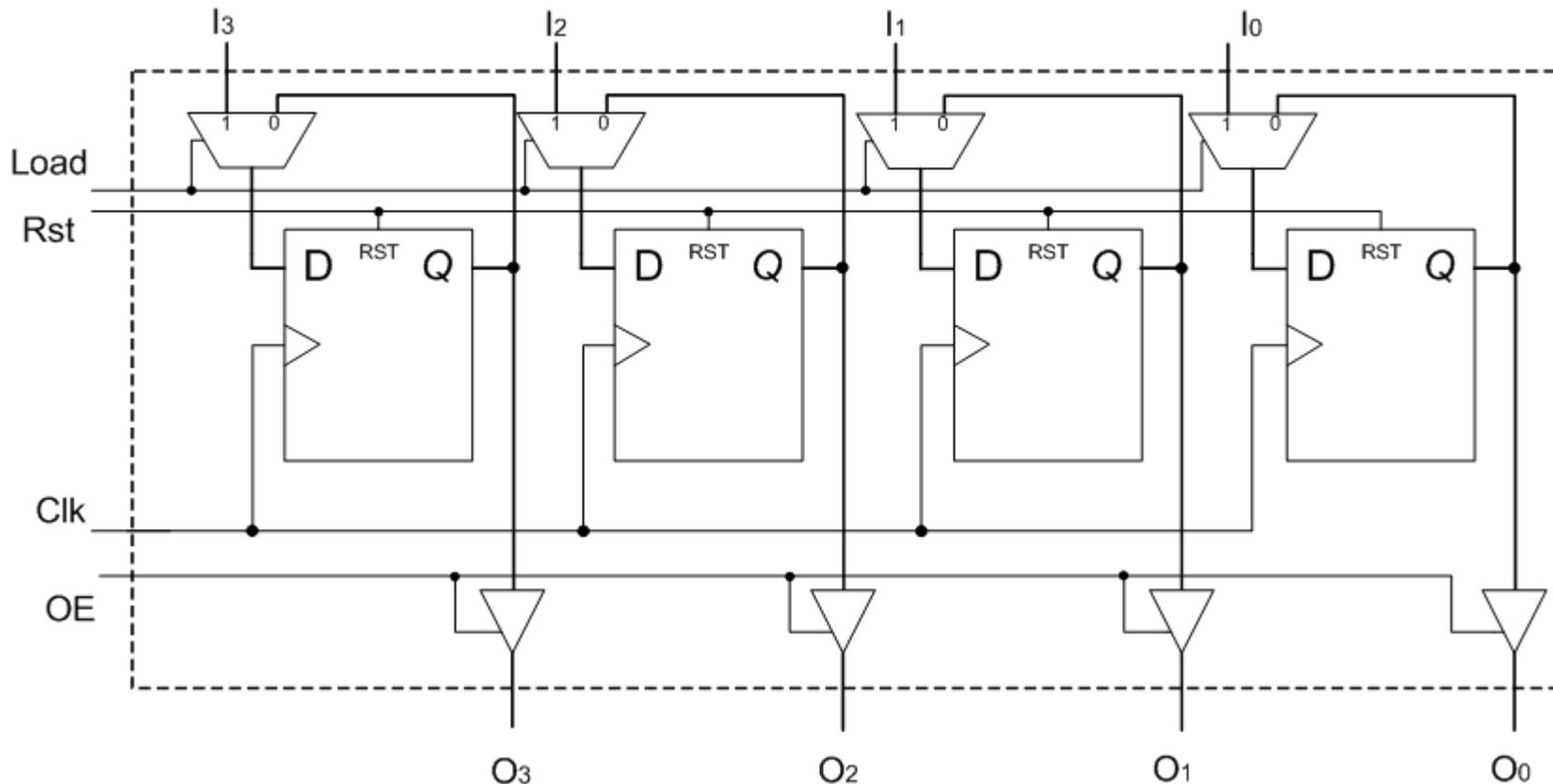
Tabla de funcionamiento: lectura

OE	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	alta impedancia			
1	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>



# REGISTROS CON CARGA PARALELA: HABILITACIÓN DE SALIDA

- Registro con carga paralela, habilitación de entrada, y habilitación de salida:





# REGISTRO CON CARGA PARALELA

- Ejemplo: modelo funcional en VHDL de un registro con carga paralela, habilitación de entrada, y habilitación de salida.

```
library ieee;
use ieee.std_logic_1164.all;

entity reg_load_oe is
  generic(width: integer);
  port(clk, rst, load, oe: in std_logic;
        I: in std_logic_vector(width-1 downto 0);
        Q: out std_logic_vector(width-1 downto 0));
end reg_load_oe;
```



# REGISTRO CON CARGA PARALELA

- Ejemplo (continuación): arquitectura.

```
architecture funcional of reg_load_oe is
  signal Q_interno: std_logic_vector(width-1 downto 0);
begin
  --Escritura:
  process(clk,rst)
    begin
      if rst = '1' then
        Q_interno <= (others => '0');
      elsif (rising_edge(clk) and load = '1') then
        Q_interno <= I;
      end if;
    end process;
  --Lectura
  process(Q_interno, oe)
    begin
      if oe = '1' then Q <= Q_interno;
      else Q <= (others => 'Z');
      end if;
    end process;
end funcional;
```



# REGISTRO CON CARGA PARALELA

- Ejemplo (continuación): test-bench de un registro de 4 bits.

```
library ieee;
use ieee.std_logic_1164.all;

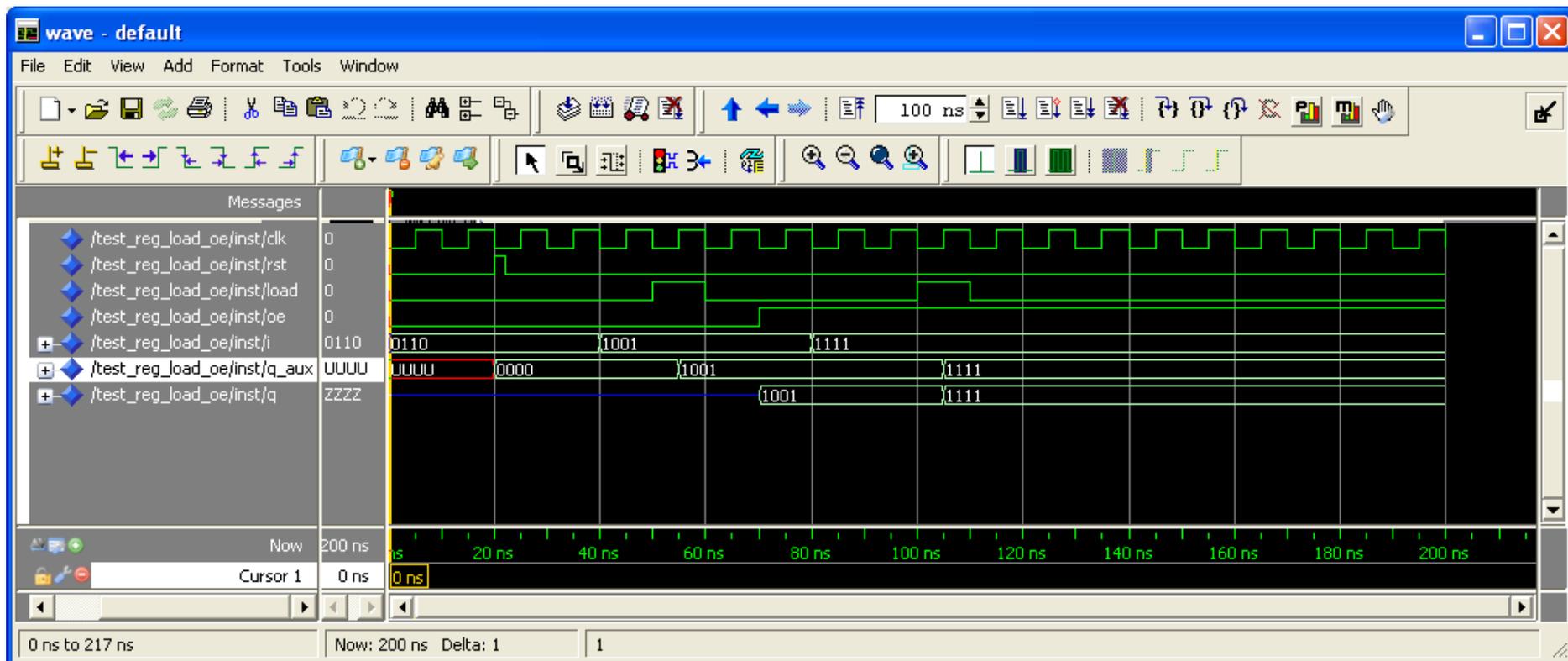
entity test_reg_load_oe is
end test_reg_load_oe;

architecture test of test_reg_load_oe is
signal clk: std_logic := '0';
signal rst, load, oe: std_logic;
signal I, Q: std_logic_vector(3 downto 0);
begin
  clk <= not clk after 5 ns;
  rst <= '0', '1' after 20 ns, '0' after 22 ns;
  load <= '0', '1' after 50 ns, '0' after 60 ns, '1' after 100 ns,
         '0' after 110 ns;
  oe <= '0', '1' after 70 ns;
  I <= "0110", "1001" after 40 ns, "1111" after 80 ns;
  inst: entity work.reg_load_oe generic map(4) port map(clk, rst, load, oe, I, Q);
end test;
```



# REGISTRO CON CARGA PARALELA

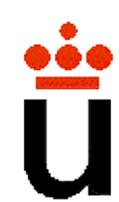
- Ejemplo (continuación): resultado de la simulación.





# ÍNDICE

- Bibliografía
- Registros
  - Carga paralela
  - Registros de desplazamiento
- Contadores
  - Contadores síncronos
  - Contadores asíncronos
- Banco de registros



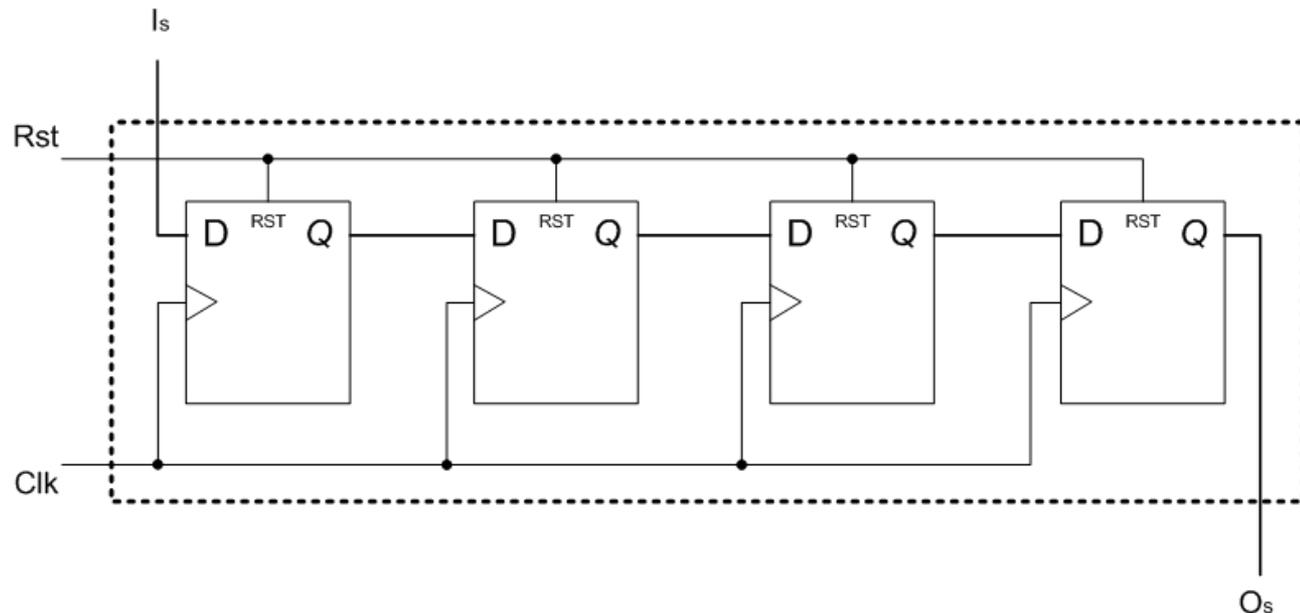
# REGISTROS DE DESPLAZAMIENTO

- Son registros que permiten realizar ciertas operaciones de carga de datos y desplazamientos a izquierda, derecha, rotaciones, etc.
- Pueden tener entradas y/o salidas de datos serie tanto por el bit más significativo como por el bit menos significativo.
- También pueden tener entradas de datos para carga en paralelo, y/o salidas en paralelo.
- En los registros de desplazamiento, su contenido se va propagando (desplazando) de un biestable a otro en cada flanco de reloj.
- Tipos de registros de desplazamiento:
  - Conversión serie / serie.
  - Conversión serie / paralelo.
  - Conversión paralelo / serie.
  - Registros de desplazamiento universales.

# REGISTRO CONVERSION SERIE/SERIE

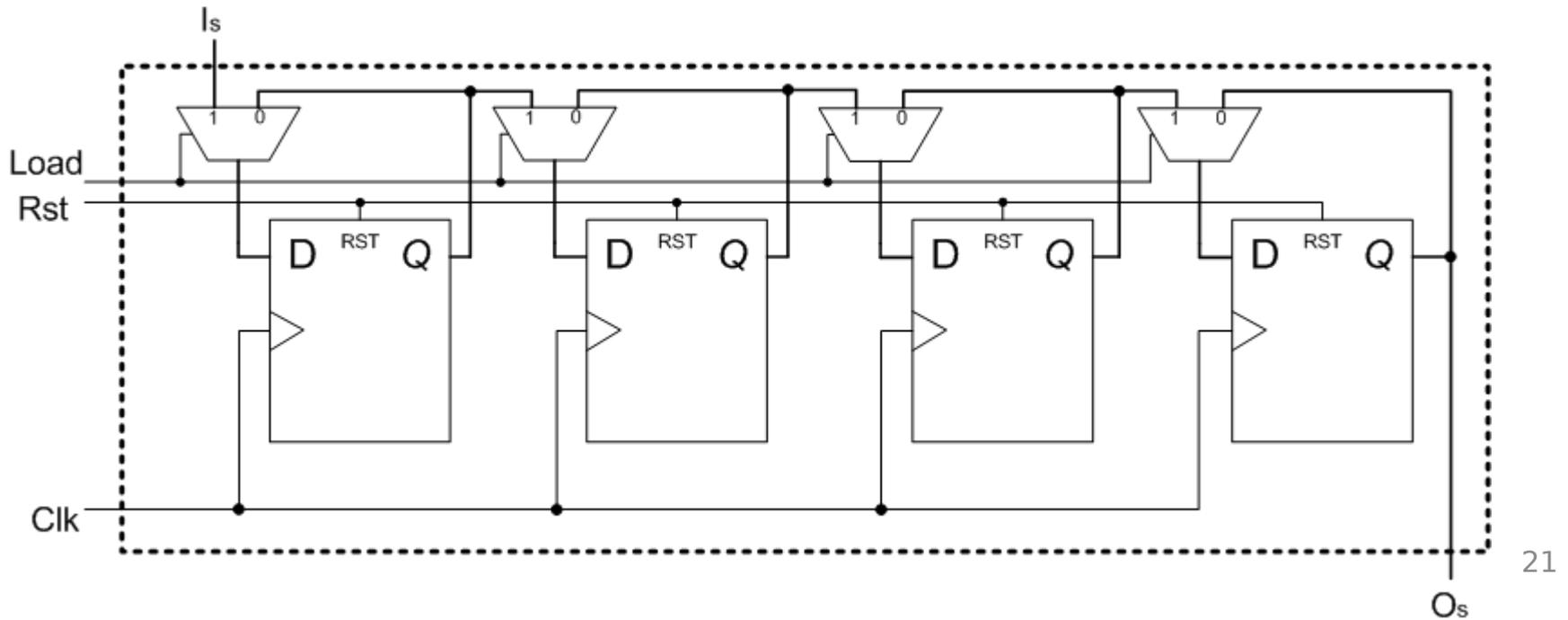
- Son registros que tienen una entrada de datos de 1 bit y una salida de datos de 1 bit. Internamente tienen 'n' biestables. El dato de entrada se va propagando (**desplazando**) de un biestable a otro en cada flanco de reloj, hasta llegar a la salida.
- La salida es igual a la entrada retrasada 'n' ciclos de reloj.

## Símbolo



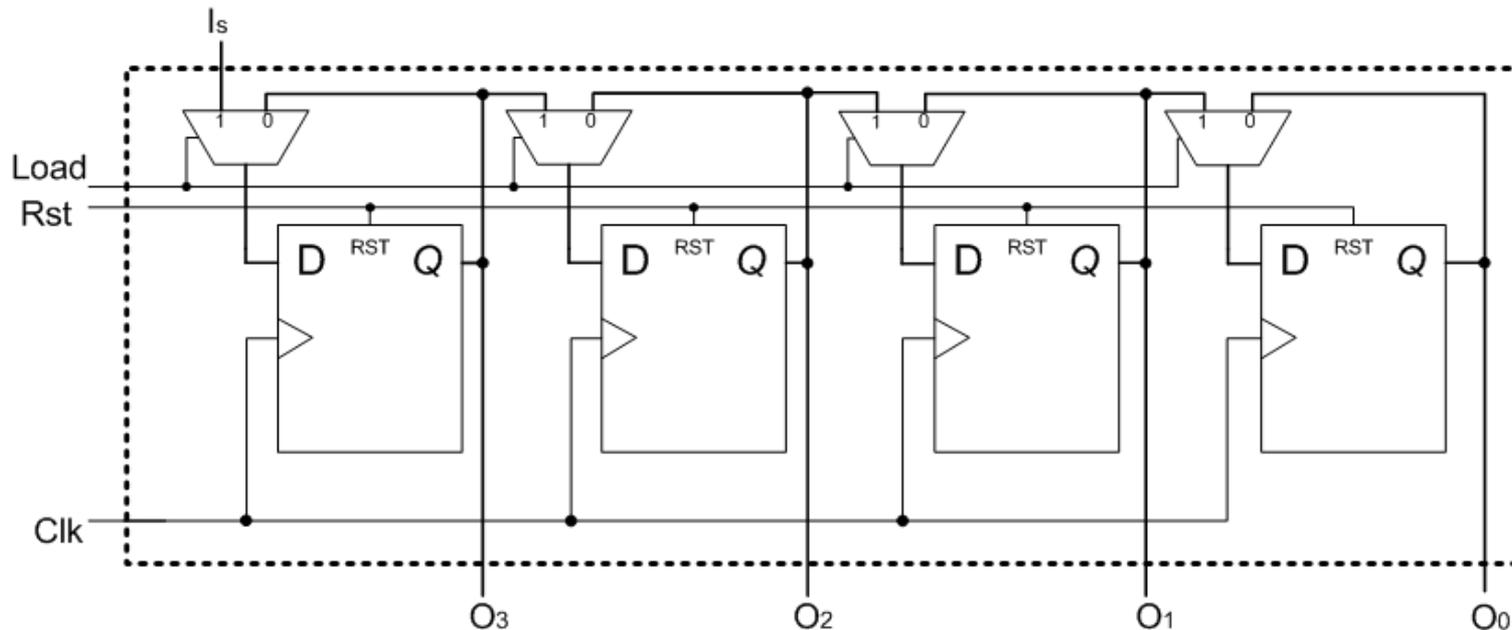
# REGISTRO CONVERSIONOR SERIE/SERIE

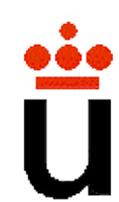
- El registro serie/serie también puede incluir entradas de habilitación de reloj, habilitación de entrada, habilitación de salida, etc.
- **Ejemplo:** registro serie/serie con habilitación de entrada y desplazamiento hacia la derecha:



# REGISTRO CONVERSION SERIE/PARALELO

- Es un registro con una entrada de datos de 1 bit y una salida de datos de 'n' bits. Internamente tiene 'n' biestables, y el dato de entrada se va propagando (**desplazando**) de uno a otro en cada flanco de reloj.
- Internamente es similar al registro serie/serie, pero todos los biestables están conectados a la salida en paralelo.
- **Ejemplo:** registro serie/paralelo con habilitación de entrada y desplazamiento hacia la derecha:





# REGISTRO CONVERTSOR PARALELO/SERIE

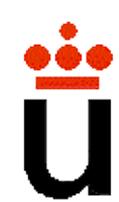
- Es un registro con una entrada de datos de 'n' bits en paralelo, y una salida de datos de 1 bit. Internamente tiene 'n' biestables, y el dato de entrada se va propagando (**desplazando**) de uno a otro en cada flanco de reloj.
- **Ejemplo:** registro paralelo/serie con habilitación de entrada y desplazamiento hacia la derecha:



# REGISTROS DE DESPLAZAMIENTO UNIVERSALES

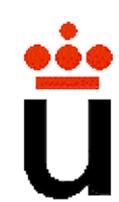
- Los registros de desplazamiento universales permiten realizar varias operaciones en función de varias señales de control:
  - Carga en paralelo.
  - Desplazamiento a la izquierda y/o a la derecha.
  - Rotación.
- Internamente están implementados utilizando multiplexores que se conectan a las entradas de los biestables y que permiten, mediante unas señales de control , escoger el tipo de operación que tiene que realizar el registro.
- Ejemplo: registro de desplazamiento con la siguiente tabla de operaciones:





# ÍNDICE

- Bibliografía
- Registros
  - Carga paralela
  - Conversión serie/serie
  - Conversión serie/paralelo
  - Registros de desplazamiento
- Contadores
  - Contadores síncronos
  - Contadores asíncronos
- Banco de registros



# CONTADORES

- Un contador es un circuito secuencial con una salida de 'n' bits que puede seguir una secuencia numérica concreta: 0, 1, 2, 3 . . .
- Los contadores pueden tener varias entradas de control:
  - Load: entrada que permite cargar un valor inicial en el contador.
  - CE: *Count Enable*. Indica si la cuenta está habilitada, o si está deshabilitada y el contador mantiene constante su valor.
  - Up/Down: entrada que indica si el contador tiene que seguir una secuencia ascendente o descendente.
- También pueden tener salidas de control como:
  - TC: (*Terminal Count*) indica que el contador está en el valor más alto (o más bajo si es descendente).
  - CEO: (*Count Enable Out*) es igual que TC pero sólo se activa si la entrada CE está activa.
- Están implementados con registros con n sumadores-restadores, y con multiplexores que manejan las distintas señales de control.
  - También pueden construirse con el método general de diseño de circuitos secuenciales (típicamente como máquinas de Moore).



# CONTADORES SÍNCRONOS

- Si los biestables comparten la señal de reloj: **contador síncrono**.
- Ejemplo: contador síncrono ascendente de 4 bits.

 Semisumador

- TC: producto de las salidas por nivel alto de los cuatro biestables.
  - En un contador descendente, TC es el producto de las cuatro salidas activas por nivel bajo.



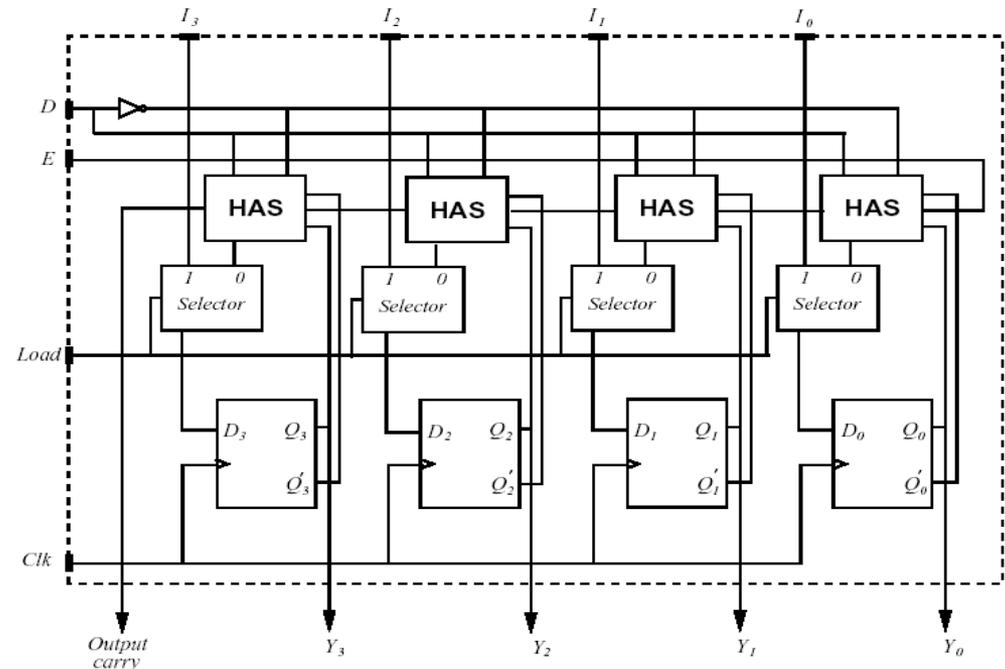
# CONTADORES SÍNCRONOS

- Ejemplo: contador síncrono ascendente/descendente de 4 bits con carga en paralelo.

HAS => Half  
Adder/Substracter

## Tabla de funcionamiento

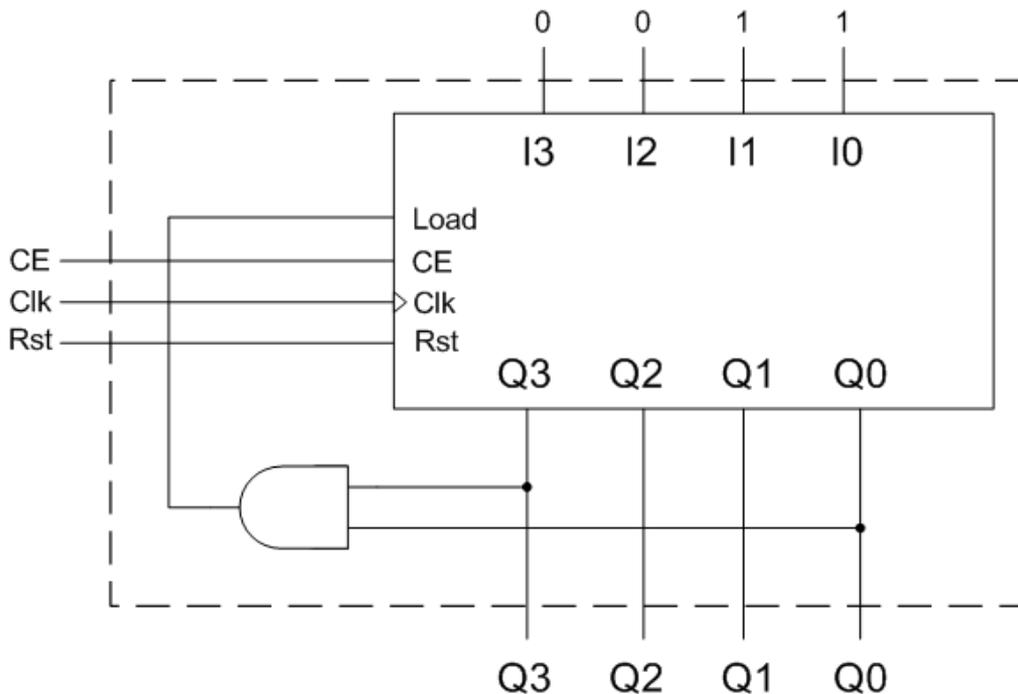
Rst	Clk	Load	E	D	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
1	X	X	X	X	0	0	0	0
0	↑	0	0	X	no cambia			
0	↑	0	1	0	Cuenta + 1			
0	↑	0	1	1	Cuenta - 1			
0	↑	0	X	X	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>



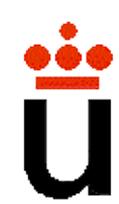
- En un contador reversible: TC = (salidas activas por nivel alto and not D) or (salidas activas por nivel bajo and D)
- CEO: siempre es el producto de TC con la entrada de habilitación de cuenta E.

# CONTADORES A-B

- Utilizando contadores con carga en paralelo se puede construir un contador que cuente desde un número inicial A hasta un número final B.
- Ejemplo: contador de 3 a 9 a partir de un contador de 0 a 15.

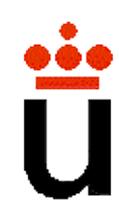


Cuando Q3 y Q0 valen '1' (en funcionamiento normal, cuando el contador contenga el valor '1001') la entrada de load se pone a '1', por lo que en el siguiente flanco de reloj se cargará en el contador el valor '0011' que está en la entrada



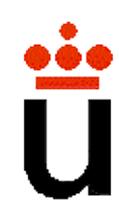
# DIVISORES DE FRECUENCIA

- Son contadores  $A-(2^n-1)$  que ofrecen a la salida únicamente la señal CEO o TC.
- El divisor de frecuencia genera una señal que tiene una frecuencia  $2^n-A$  veces menor que el reloj que reciben
- Ejemplo: divisor de frecuencia que genera una señal de reloj  $\text{Clk}'$  de frecuencia  $1/10$  de la del reloj original  $\text{Clk}$  partiendo de un contador de 0 a 15.



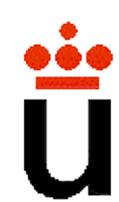
# AGRUPACIÓN DE CONTADORES SÍNCRONOS

- Se pueden construir contadores síncronos más grandes agrupando contadores pequeños, manejando apropiadamente las entradas de habilitación de los contadores y las salidas de fin de cuenta.
- Ejemplo: contador de 8 bits (0 a 255) a partir de 2 contadores de 4 bits (0 a 15).



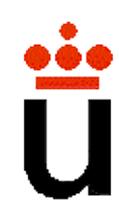
# CONTADORES ASÍNCRONOS

- Se les llama también contadores por propagación o *ripple counters*.
- En ellos los biestables no comparten la señal de reloj.
- Los biestables no transitan simultáneamente, sino que las transiciones se van propagando de unos biestables a otros.
  - Al no bascular todos los biestables a la vez, se producen estados espurios con valores incorrectos, y la frecuencia de reloj del contador está limitada.
- Ejemplo: contador asíncrono de 4 bits.



# ÍNDICE

- Bibliografía
- Registros
  - Carga paralela
  - Conversión serie/serie
  - Conversión serie/paralelo
  - Registros de desplazamiento
- Contadores
  - Contadores síncronos
  - Contadores asíncronos
- Banco de registros



# BANCO DE REGISTROS

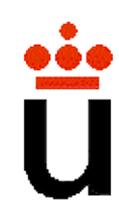
- Un banco de registros es un conjunto de registros síncronos por flanco que comparten las líneas de entrada y las líneas de salida, y que mediante unos decodificadores de lectura y de escritura permiten seleccionar qué registro se lee y en qué registro se escribe.
- Ejemplo: banco de registros con 4 registros de 4 bits.

WE (write enable): controla si se permite o no la escritura en el banco.

WA (write address): controla en cuál de los registros se escribirá en el próximo flanco de reloj.

RE (read enable): controla si se permite o no la lectura del banco (no suele ser necesaria).

RA (read address): controla que registro del banco se lee.



# BANCO DE REGISTROS

- Ejemplo (continuación): implementación con registros con habilitación de carga y de salida, y decodificadores.

Por claridad se han obviado las señales de reloj y de *reset*.

Este banco tiene un **puerto de salida** (lectura) y un **puerto de entrada** (escritura).

Es frecuente que los bancos de registros tengan **varios puertos de salida** para permitir leer varios registros a la vez.